

2 / 14

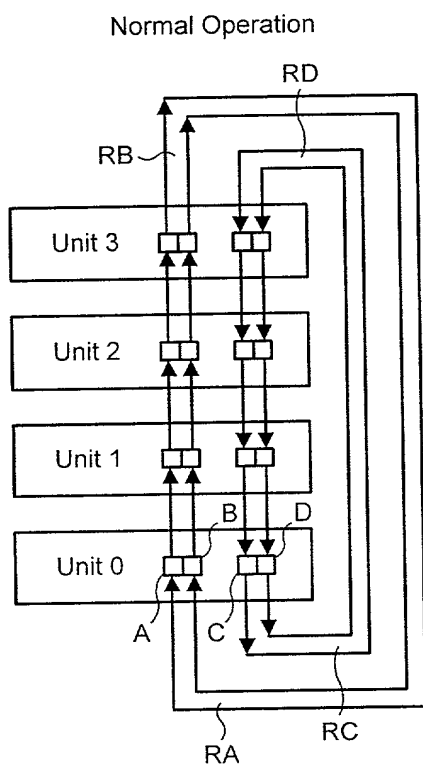


FIG. 2

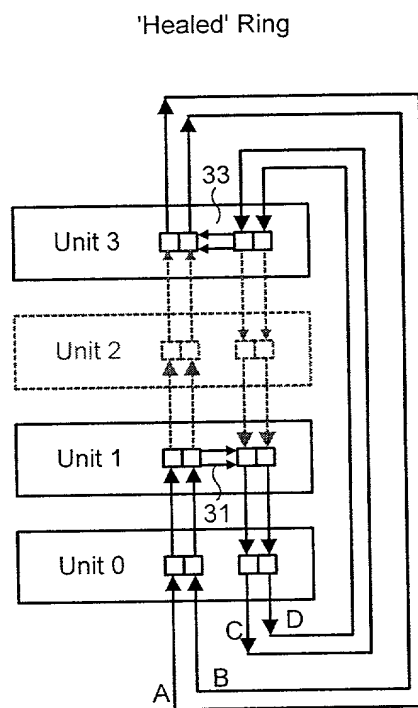


FIG. 3

3 / 14

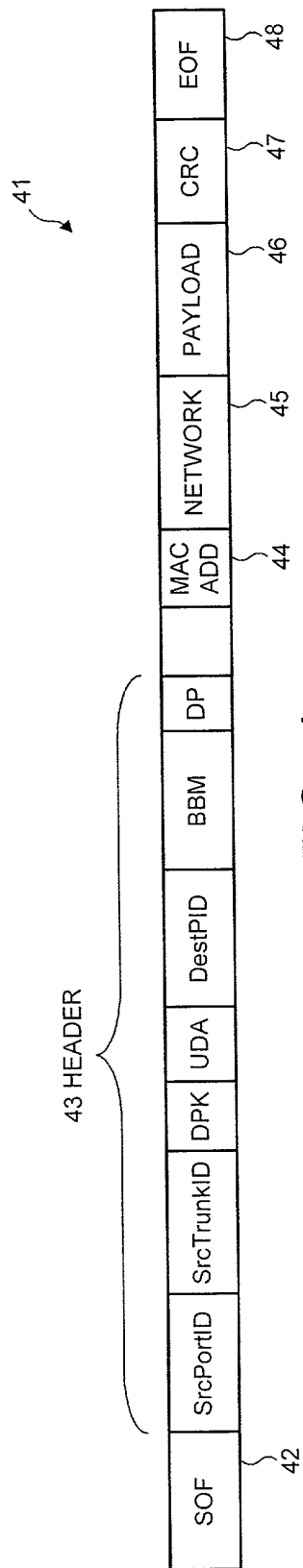


FIG. 4

4 / 14

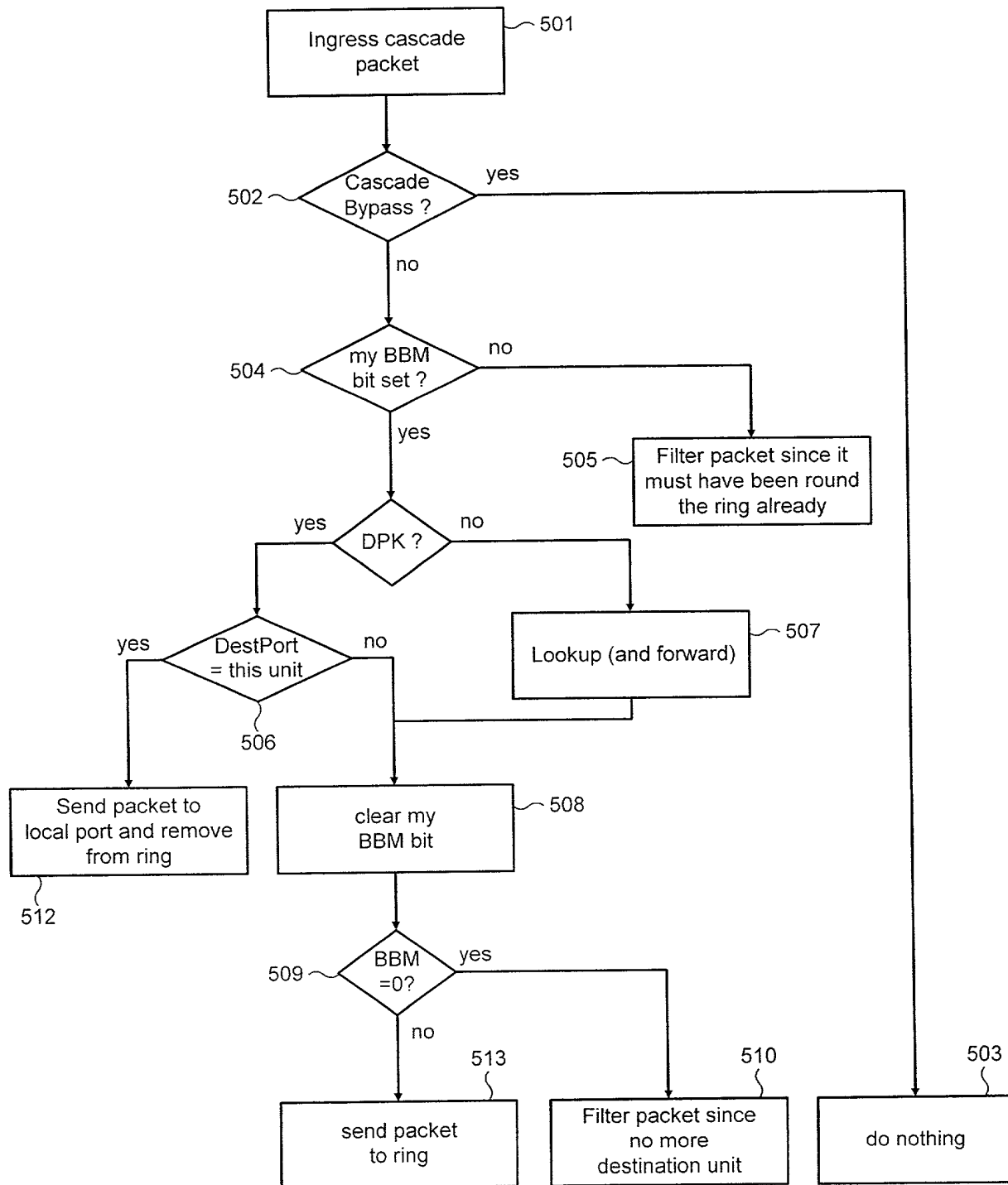


FIG. 5

5 / 14

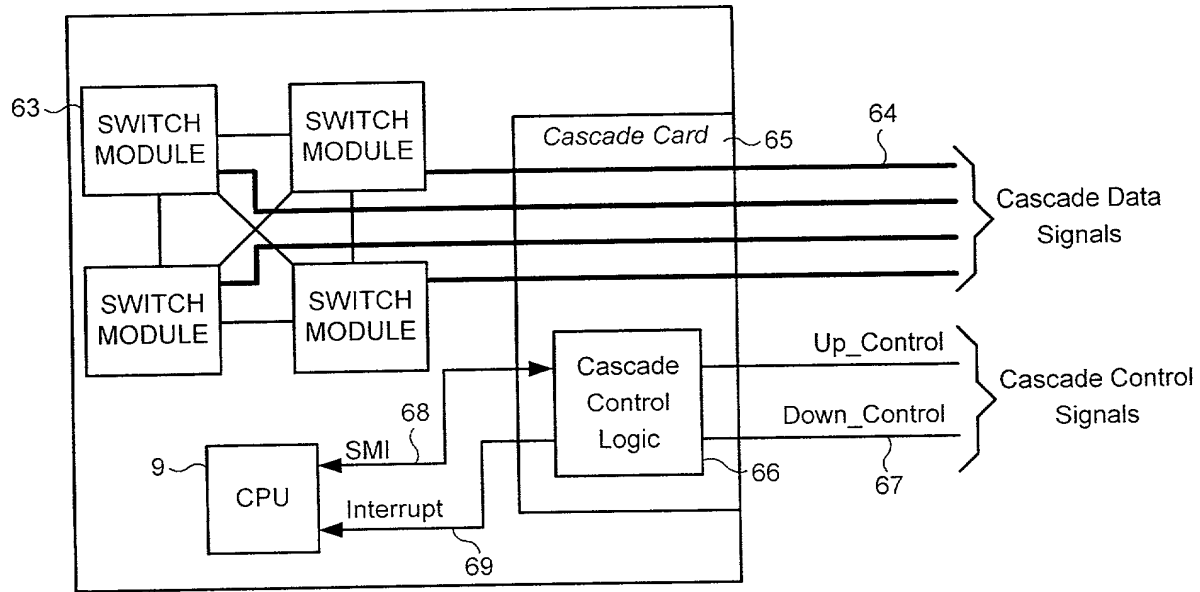


FIG. 6

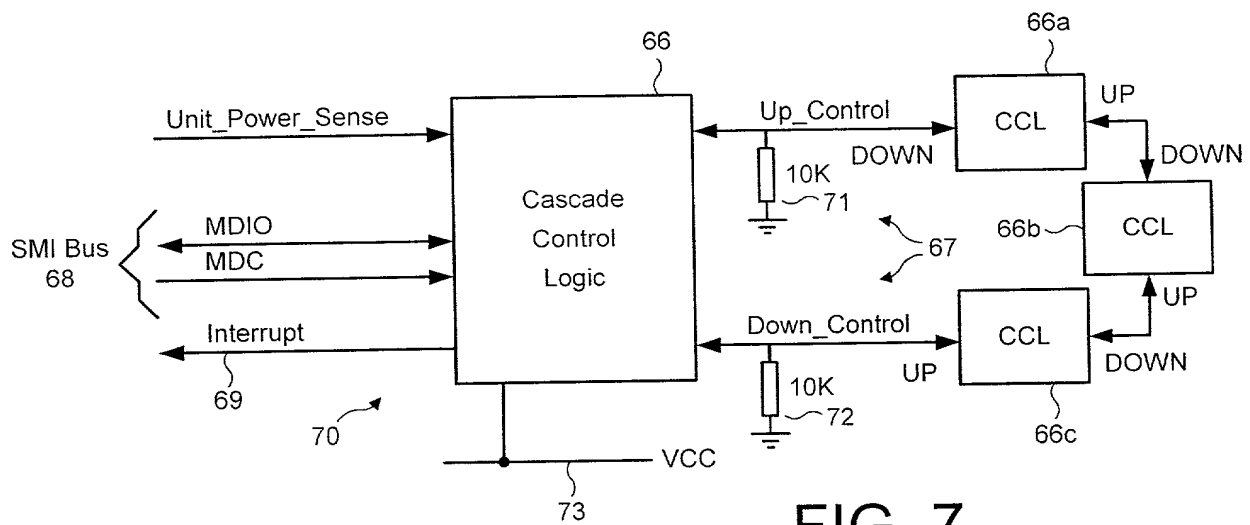


FIG. 7



7 / 14

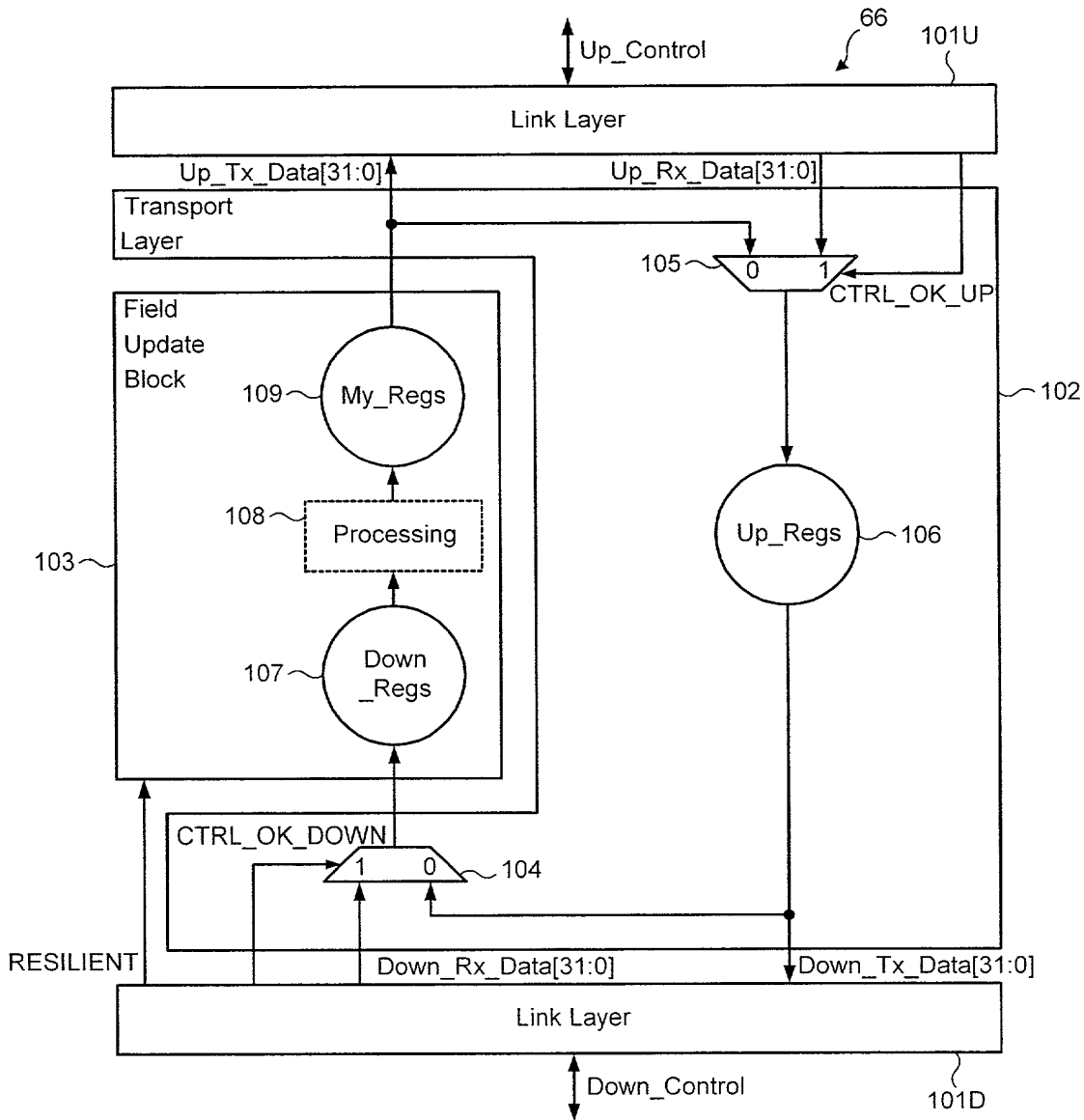


FIG. 10

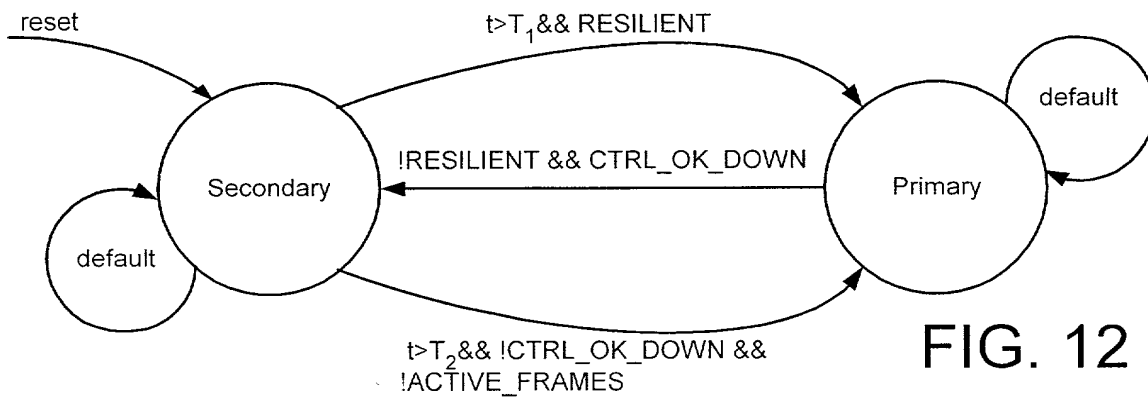


FIG. 12

8 / 14

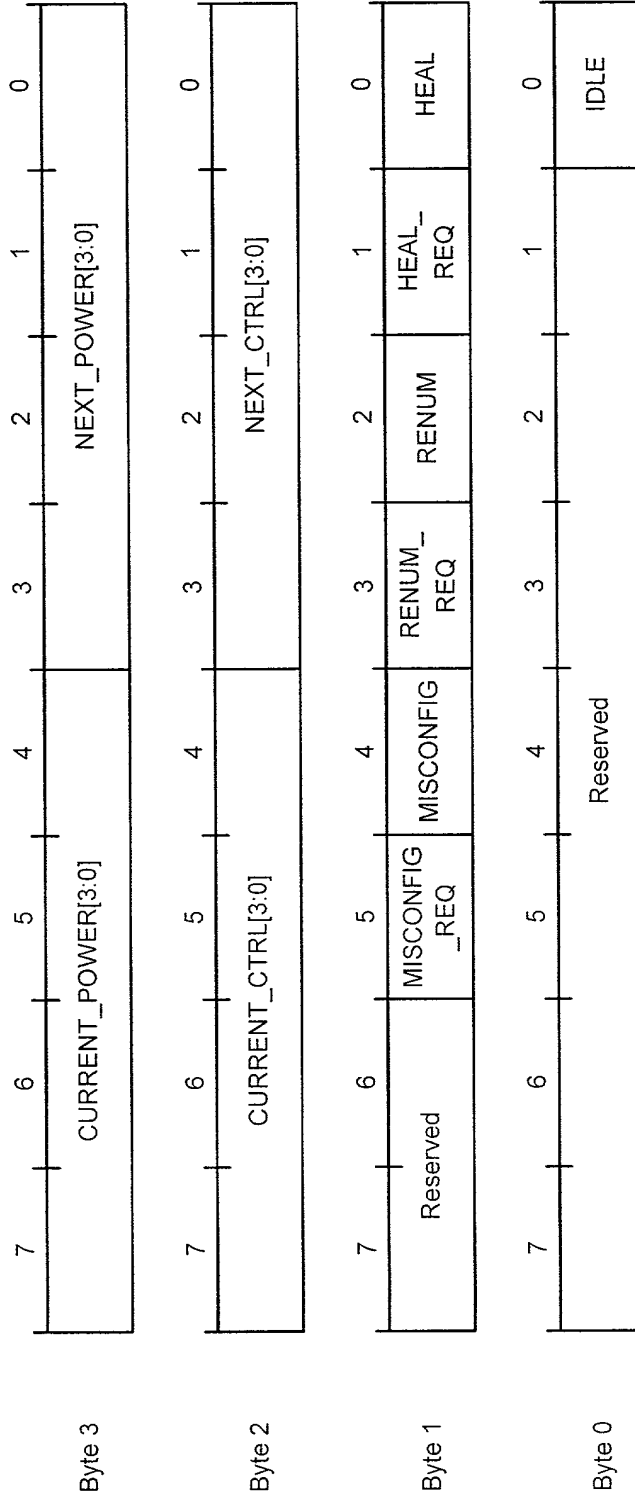


FIG. 11



9 / 14

```
always @(posedge clk or negedge reset)
begin
  if (reset) begin
    my_regs_UNIT_ID <= 0;
    my_regs_MISCONFIG_REQ <= FALSE;
    my_regs_RENUM_REQ <= FALSE;
  end
  else begin
    my_regs_UNIT_ID <= my_regs_UNIT_ID;
    my_regs_MISCONFIG_REQ <= FALSE;
    my_regs_RENUM_REQ <= FALSE;

    if (PRIMARY) begin
      if (RESILENT)
        my_regs_UNIT_ID <= 0;
      end
      else begin
        if (!down_regs_IDLE) begin
          if (down_regs_UNIT_ID==3 &&
down_regs_NEXT_CTRL==4'b1111)
            my_regs_MISCONFIG_REQ <= TRUE;
          else if (down_regs_UNIT_ID == 3)
            my_regs_RENUM_REQ <= TRUE;
          else
            my_regs_UNIT_ID <= down_regs_UNIT_ID + 1;
          end
        end
      end
    end
  end
end

always
begin // generate UNIT_ID from NEXT_CTRL
  casex (down_regs_NEXT_CTRL)
    4'b1xxx: down_regs_UNIT_ID = 3;
    4'b01xx: down_regs_UNIT_ID = 2;
    4'b001x: down_regs_UNIT_ID = 1;
    default: down_regs_UNIT_ID = 0;
  endcase
end
```

FIG. 13

10 / 14

```
always @(posedge clk or negedge reset)
begin
  if (reset) begin
    my_regs_CURRENT_POWER <= 4'b0000;
    my_regs_NEXT_POWER    <= 4'b0000;
    my_regs_CURRENT_CTRL  <= 4'b0000;
    my_regs_NEXT_CTRL     <= 4'b0000;
  end
  else begin
    if (!down_regs_IDLE) begin
      if (PRIMARY) begin
        my_regs_CURRENT_POWER <=
          down_regs_NEXT_POWER;
        my_regs_NEXT_POWER    <= MY_POWER <<
          my_regs_UNIT_ID;
        // MY_POWER is a sample of the input
        Unit_Power_Sense
        my_regs_CURRENT_CTRL <=
          down_regs_NEXT_CTRL;
        my_regs_NEXT_CTRL <= 1'b1 <<
          my_regs_UNIT_ID;
      end
      else begin
        my_regs_CURRENT_POWER <=
down_regs_CURRENT_POWER;
        my_regs_NEXT_POWER <=
          down_regs_NEXT_POWER | (MY_POWER <<
            my_regs_UNIT_ID);
        my_regs_CURRENT_CTRL <=
          down_regs_CURRENT_CTRL;
        my_regs_NEXT_CTRL <=
          down_regs_NEXT_CTRL | (1'b1 <<
            my_regs_UNIT_ID);
      end
    end
    else begin
      // leave registers unchanged
    end
  end
end
```

FIG. 14

|        |                    |               |           |                 |       |          |
|--------|--------------------|---------------|-----------|-----------------|-------|----------|
| Byte 3 | CURRENT_POWER[3:0] |               |           | NEXT_POWER[3:0] |       |          |
| Byte 2 | CURRENT_CTRL[3:0]  |               |           | NEXT_CTRL[3:0]  |       |          |
| Byte 1 | Reserved           | MISCONFIG_REQ | MISCONFIG | RENUM_REQ       | RENUM | HEAL_REQ |
| Byte 0 | Reserved           |               |           | Reserved        |       |          |
|        | 7                  | 6             | 5         | 4               | 3     | 2        |
|        |                    |               |           |                 |       | 1        |
|        |                    |               |           |                 |       | 0        |

FIG. 15

|        |                    |               |           |                 |            |              |              |
|--------|--------------------|---------------|-----------|-----------------|------------|--------------|--------------|
| Byte 4 | ACTIVE_FRAMES      | MY_POWER      | RESILIENT | PRIMARY         | CTRL_OK_UP | CTRL_OK_DOWN | UNIT_ID[1:0] |
| Byte 3 | CURRENT_POWER[3:0] |               |           | NEXT_POWER[3:0] |            |              |              |
| Byte 2 | CURRENT_CTRL[3:0]  |               |           | NEXT_CTRL[3:0]  |            |              |              |
| Byte 1 | Reserved           | MISCONFIG_REQ | MISCONFIG | RENUM_REQ       | RENUM      | HEAL_REQ     | HEAL         |
| Byte 0 | Reserved           |               |           | Reserved        |            |              | IDLE         |
|        | 7                  | 6             | 5         | 4               | 3          | 2            | 1            |
|        |                    |               |           |                 |            |              | 0            |

FIG. 16

FIG. 17

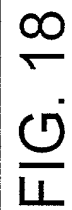


FIG. 18

## CCL Register Fields in Normal Mode

|        | CURRENT_<br>POWER | CURRENT_<br>CTRL | HEAL | CTRL_OK_UP | CTRL_OK_<br>DOWN |
|--------|-------------------|------------------|------|------------|------------------|
| Unit-3 | 1011              | 1111             | 0    | 1          | 1                |
| Unit-2 | 1011              | 1111             | 0    | 1          | 1                |
| Unit-1 | 1011              | 1111             | 0    | 1          | 1                |
| Unit-0 | 1011              | 1111             | 0    | 1          | 1                |

## CCL Register Fields in Healed Mode

FIG. 20

### CCL Register Fields in Healed Mode (due to missing cable)

FIG. 21